User's Guide: 1-D Radiative Transfer Code

Document version: 1.0 June 02, 2011

James Schaeffer

Compiled with LaTeX: June 2, 2011

Contents

Chapter 1

Introduction

This document describes how to compile and run the 1-D radiation transfer program, driver.f. This is the radiative transfer code used in the NASA Ames Mars general circulation model (v23, September 2010).

1.1 Radiative Transfer: Overview

The radiative transfer (RT) code was taken from a 3-D general circulation model (GCM) that calculates fluxes in the atmosphere at layer boundaries, and at the surface. GCMs typically use these fluxes to calculate temperatures from heating rates in the model atmosphere, as well as the amount of solar energy absorbed by the surface. In order to calculate the fluxes, the radiative transfer code uses GCM variables such as layer temperatures, pressures, and dust loading, which are used to calculate optical depths in each layer. Variables that a 3-D GCM would compute and then pass to the radiation code are set by the user in driver.f, and then passed to the radiative transfer subroutines as needed. The results of the radiative transfer calculations are fluxes at layer boundaries. Heating rates are calculated from the fluxes. These heating rates are computed near the end of driver.f and written to the output file, driver_data.

The radiation transfer code is based on the two-stream approximation; quadrature in the visible, hemispheric mean in the IR. This aspect of the code is not designed to be changed. However, such things as the number of spectral intervals, the dust properties, or the gas species in the atmosphere can be modified. The spectral intervals, the absorbing gasses, the aerosol optical properties, and other model properties, are specified by the user, and are easily changed as needed.

1.2 Radiative Transfer: Current Model

The RT code includes CO_2 and H_2O gas opacity as well as aerosol optical properties for dust, and water ice clouds. Non-RT physics computed in a 3-D GCM is provided by the user in driver.f. These include cosine of the solar zenith angle, dust optical depth and temperature in each layer, to name a few quantities. The RT code computes the visible and IR fluxes, as well as net fluxes, at the layer boundaries. Heating rates can be computed from these fluxes. Currently the code is set up to calculate the flux through the specified atmosphere for a single time step. The code can be modified to run for multiple time steps: through a diurnal cycle, or run until radiative equilibrium is reached.

The radiation code uses correlated-k gas opacities. For each spectral interval there are 17 Gauss points. The radiative transfer equation is solved throughout the entire atmospheric column for each Gauss point in each spectral interval. The current setup uses 12 spectral intervals (7 visible, 5 IR), with 17 Gauss points per interval.

The RT code uses different two-stream approximations for the visible and IR spectral regions, and computes the visible and IR fluxes separately. The result is that fluxes in each spectral interval are calculated, and combined into values for the entire spectral region: one for the visible, another for the IR. These represent fluxes for the entire visible spectral region and a separate set of values for the entire IR region. Net fluxes at the boundary of each layer for the visible and IR regions are computed from these fluxes. The returned values from the radiation code are visible and IR fluxes (and net flux) at layer boundaries, which can be turned into heating rates. The net solar flux at the surface is also computed in the visible band.

Chapter 2

Quick Start

2.1 Some Information

Download the file rtdriver.tar.gz and place it in a working directory. Change to that directory and unpack the tar file:

```
Mars> tar zxvf rtdriver.tar.gz
```

where "Mars>" (and "IDL>" below) represent shell prompts. This should create a directory rtdriver which holds the radiation source code. To compile the code go to the directory rtdriver and run the make file:

```
Mars> cd rtdriver
Mars> make
```

If all went well, the program driver should have been created. To run the code, create and view a sample plot, (assuming IDL is installed):

```
Mars> driver
Mars> idl
IDL> driver
IDL> exit
Mars> gv driver.ps
```

where gv is any program that displays postscript files. If IDL is not installed, a graphics program of your choice can be used. The RT program driver outputs the file, driver_data that holds the output values used for plotting. The file driver.pro is an IDL script that reads the output file and creates the IDL plot.

2.2 A Little More Information

The code and data files are setup to run on standard little-endian (x86, x86_64) Linux computers using the GFORTRAN compiler. The rtdriver distribution is designed to run with no modifications on such Linux computers. The downloaded file, rtdriver.tar.gz expands into a directory rtdriver, which contains the source code, and a directory, data. This directory includes files that hold the CO_2 and H_2O absorption coefficients, as well as dust and water ice cloud optical constants.

The code has been compiled using various FORTRAN90 compilers, but this distribution uses GFORTRAN as the default. GFORTRAN is part of the standard Fedora distribution, as well as other Linux configurations, and can be installed on machines using different operating systems. The code needs GFORTRAN version 4.3 or later to work properly. The default make file (Makefile) uses GFORTRAN. This file can be modified as needed if a different compiler is desired.

Chapter 3

More About the Code

3.1 What files are needed?

In addition to the source code and Makefile, there are four input data files which are required to run the code: CO2H2O_V_12_95 and CO2H2O_IR_12_95, as well as QEXT_DUST and QEXT_WATER. These are the visible and IR k-coefficient data files for CO_2 (plus various amounts of H_2O) gas absorption, and the optical constants for both dust and water ice aerosols. Below is a table which lists the set of required data files, showing their names, which subroutine reads each file, and data type.

Description	File Name	Subroutine	Data Type
Gas Opacity			
k-coefficients (V)	CO2H2O_V_12_95_INTEL	laginterp.f90	real*8
k-coefficients (IR)	CO2H2O_IR_12_95_INTEL	laginterp.f90	real*8
Optical Constants			
Dust	QEXT_DUST	setrad.f90	ASCII
Water Ice	QEXT_WATER	setrad.f90	ASCII

Table 3.1: Radiation transfer data files.

3.2 Pressure and Temperature Limits

The gas absorption k-coefficients were calculated off-line on a pressure (P) and temperature (T) grid. The pressure ranges from 10^{-6} to 10^{+4} mbar, with the pressure at each grid point 10 times larger than the previous one. The temperature grid ranges from 50 K to 350 K, $\Delta T = 50$ K.

For given values of P and T, the code interpolates the k-coefficients from the above described grid to the requested P,T values. If either P or T is outside the limits of the grid, the code uses the appropriate limiting value (e.g. if $T = 360$ K then the code uses the k-coefficients for $T = 350$ K).

The variable, PTROP (in driver.f) is set to 2.0×10^{-6} mbar. The code uses $PTROP/2$ as the top of the atmosphere, which in this case corresponds to the limit of valid pressures for the k-coefficient grid. To stay within the pressure limits of the code, select PTROP

such that $PTROP \geq 2.0 \times 10^{-6}$ mbar. To stay within the pressure limits keep the surface pressure, PSF, within the values $PTROP < PSF \leq 10^{+4}$ mbars.

3.3 Dust Optical Properties

The dust optical properties, Q_{ext} (extinction efficiency), Q_{scat} (scattering efficiency), and g (scattering asymmetry parameter), are calculated off-line and values read in the code in setrad.f90. Note that the aerosol data are listed in increasing wave number, not wavelength. The radiation code variables are QEXTV, QSCATV, GV (in the visible) and QEXTI, QSCATI, GI (in the IR). The spectral intervals are shown in the appendix.

3.4 User Defined Variables

There is no user input file, just various variables in driver.f that can be set. This includes the temperature, pressure, dust loading, and water mixing ratio at each layer. Temperature profiles, as well as dust optical depth profiles can be included at the compile stage. There are examples in driver.f of both temperature and dust opacity files being included in this manner.

Below is a short list of variables in driver.f that the user may wish to modify. Units are MKS except for pressures, which are in mbar, and the (square) of the Sun-Mars distance, which is in astronomical units, AU.

ALBI: Surface albedo in the IR.

ALBV: Surface albedo in the visible.

PTROP: Tropopause pressure, mbar.

PSF: Surface pressure, mbar.

TAUTOT: Visible dust optical depth at the reference pressure level, RPTAU. RPTAU is set to the surface pressure in this version of the code.

CONRNU: Sets vertical dust mixing; 10^{-8} for uniformly mixed dust, 0.3 for dust confined to the lowest model layers.

ACOSZ: Cosine of the solar zenith angle.

QH2O: Water mixing fraction. The limits are 10^{-7} to 0.4. Set to 10^{-7} for no water.

RSDIST: Square of the Sun-Mars distance, in $(AU)^2$.

GRAV: Acceleration due to gravity (3.72 m/s^2).

GCMLAYERS: Number of model layers.

TSTRAT: Temperature of the stratosphere.

TL: Atmosphere temperature at each level.

GT: Ground temperature.

DSIG: Layer thickness on the σ-coordinate system.

SIGMA: Vertical coordinate system. $\sigma = 0$ at the tropopause, $\sigma = 1$ at the surface.

TAUREF: Dust optical depth of each sub-layer, at the reference wavelength.

TAUCUM: Cumulative dust optical depth measured from the top of the atmosphere to each level.

PLEV: Pressure at the GCM model levels.

TLEV: Temperature at the GCM model levels.

SOL: Solar flux in each spectral interval, at the Mars-Sun distance.

3.5 Code Flow

An overview of the RT code flow is presented below. This is a high level description, based on how the code is used in a 3-D model. No changes to the overall flow where made to take into account that this is a 1-D driver. A short description of what each subroutine does is given in the appendix.

INITIALIZATION

(1) Initialize local variables and input variables, such as the surface albedo and τ_d (the dust optical depth).

(2) CALL RADSETUP: This initializes the radiation code. The spectral intervals are set in the visible (setspv.f90) and IR (setspi.f90). The dust optical properties Q_{ext}, Q_{scat}, and g are set in setrad.f90.

(3) User defined variables, such as surface pressure, tropopause pressure, and cosine of the solar zenith angle, are set. The atmosphere temperature is set, either directly, or by including a file that lists the values at each pressure level. The ground temperature is set. The pressure at each level is computed. The dust profile is either computed (CALL DUSTPROFILE) or placed into the program at compile time.

(4) Put P, T into GCM-like arrays PLEV, TLEV, as well as radiation code arrays PMID, TMID.

(5) Calculate the Solar flux at the top of the atmosphere in each visible spectral interval for the requested Mars-Sun distance (RSDIST).

RADIATION CALCULATIONS

(6) If the sun is up then calculate the visible optical depth (optcv.f90), and the fluxes in the solar spectral intervals (sfluxv.f90).

(7) If the sun is down ($ACOSZ < 1.0 \times 10^{-4}$), set the visible fluxes to zero.

(8) Calculate the IR optical depth (optci.f90), and the fluxes in the IR spectral intervals (sfluxi.f90).

(9) The fluxes have been computed: radiative transfer is done. FMNETV and FMNETI are the net visible and IR fluxes. FLUXUPV and FLUXUPI are the upward visible and IR fluxes. FLUXDNV and FLUXDNI are the downward visible and IR fluxes. Note, that while heating rates are the primary reason for doing radiative transfer in GCMs, they are a derived quantity (calculated using net fluxes), and as such not listed as part of this RT code.

OUTPUT

(10) Output various quantities: fluxes, heating rates ....

Chapter 4

Appendix

4.1 Spectral Intervals

There are 12 spectral intervals in the current version of the code: seven in the visible, five in the IR. Table 4.1 below lists the spectral intervals in the visible, table 4.2 shows the IR spectral intervals. The solar flux (at 1 AU) inside each band is shown in the first table. The total solar flux is 1356.1 W/m^2.

Table 4.1: Visible spectral intervals and solar flux at 1 AU.

GCM Band		Wavelength Interval (μm)	$F_{Solar}(W/m^2)$
1		4.50 – 3.24	12.7
2		3.24 – 2.48	24.2
3		2.48 – 1.86	54.6
4		1.86 – 1.31	145.9
5		1.31 – 0.80	354.9
6		0.80 – 0.40	657.5
7		0.40 – 0.24	106.3

Note that the reference wavelength in the visible, $0.67\mu m$, corresponds to GCM visible band #6. The IR reference wavelength, $9\mu m$, corresponds to GCM IR band #4.

Table 4.2: IR spectral intervals.

GCM Band		Wavelength Interval (μm)
1		1000 – 60.0
2		60.0 – 24.0
3		24.0 – 12.0
4		12.0 – 8.00
5		8.00 – 4.50

4.2 Subroutines

A short description of the radiation subroutines is given. The subroutines are listed in the order in which they're called.

DRIVER — This is the program which sets up the variables and then calls the various radiation subroutines. Output, such as heating rates, is calculated here.

INITIALIZATION

RADSETUP — This is subroutine calls SETSPV, SETSPI, and SETRAD. It also sets the reference Q_{ext} value. This information is used to scale the IR Q_{ext} and Q_{scat} values.

SETSPV — The visible spectral intervals are defined. The solar flux (at 1 AU) is set in each spectral interval. Finally, the P,T-independent part of Rayleigh scattering is computed.

SETSPI — The IR spectral intervals are defined. The Planck integral for each spectral interval is computed for a fine grid of temperatures, ranging from 50 K to 900 K.

SETRAD — The aerosol optical properties (Q_{ext}, Q_{scat}, and g) for both visible and IR are set. The reference P and T grids are also defined. These are the pressure and temperatures at which the CO_2 + H_2O k-coefficients were calculated. Subroutine LAGINTERP is called, which reads in the k-coefficients.

LAGINTERP — The visible and IR k-coefficients are read in. The fine-mesh pressure grid is set here. The k-coefficients are interpolated to this pressure grid using Lagrange interpolation. LAGRANGE is called to do this. This interpolation allows a simple 4-point linear interpolation (in T and P) of the k-coefficients to be performed, saving time.

LAGRANGE — Performs a Lagrange interpolation of the k-coefficients as a function of pressure.

DUSTPROFILE — This fills the dust optical depth array with a user-defined dust profile.

CALCULATION

OPTCV — Computes the visible optical properties throughout the column. These are due to CO_2 and Rayleigh scattering, and when present, H_2O, dust, and clouds. Calculates TAUGSURF, the gas optical depth from the top of the atmosphere to the surface. Returns τ^v, Q^v_{ext}, ω^v_o, and g^v.

OPTCI — Computes the IR optical properties, throughout the column. These are due to CO_2 and when present, H_2O, dust, and clouds. Calculates TAUGSURFI, the gas optical depth from the top of the atmosphere to the surface. Returns τ^{ir}, Q^{ir}_{ext}, ω^{ir}_o, and g^{ir}.

TPINDEX — Finds the indices in the reference P, T, and $q(H_2O)$ grids needed to interpolate to the requested P, T, and $q(H_2O)$ values of the layer in question.

SFLUXV — Sets up the boundary conditions at the surface and top of the atmosphere for each spectral interval and Gauss point. It calls GFLUXV to solve the radiative transfer equation. It sums the results from each Gauss point and spectral interval into a single flux at each atmosphere layer. SFLUXV returns visible fluxes.

SFLUXI — Sets up the boundary conditions at the surface and top of the atmosphere for each spectral interval and Gauss point. It calls GFLUXI to solve the radiative transfer equation. It sums the results from each Gauss point and spectral interval into a single flux at each atmosphere layer. SFLUXI returns IR fluxes.

GFLUXV — Solves the radiative transfer equation for visible fluxes throughout the column. It first scales τ, ω_o, and g to take into account the strongly forward peaked scattering phase function. It then puts the appropriate values into tridiagonal form, and calls DSOLVER to solve the equations. Fluxes are returned

GFLUXI — Solves the radiative transfer equation for IR fluxes throughout the column. It puts the appropriate values into tridiagonal form, and calls DSOLVER to solve the equations. Fluxes are returned.

DSOLVER — Solves for the coefficients of the two stream solution. Calls DTRIDGL.

DTRIDGL — Solves a system of tridiagonal matrix equations.

4.3 Vertical Grid

The radiation code vertical grid has layer 1 at the top. In our case layer 25 is at the surface. The RT boundary conditions are applied at the *Top* level (see figure 4.1), where there is no downward IR flux, and the solar flux is the direct solar flux at the Sun-Mars distance. The bottom boundary conditions, applied at the *Surface* level, are that the upward IR flux is σT_g^4, where T_g is the surface temperature, and the upward solar flux is the reflected part of the downward beam.

4.4 Sample Output

The unmodified version of driver.f (the baseline code) produces output shown in figure 4.2. The input parameters are listed in the *Net Solar Flux* plot. The plot was made using driver.pro, which uses driver_data as the data file.

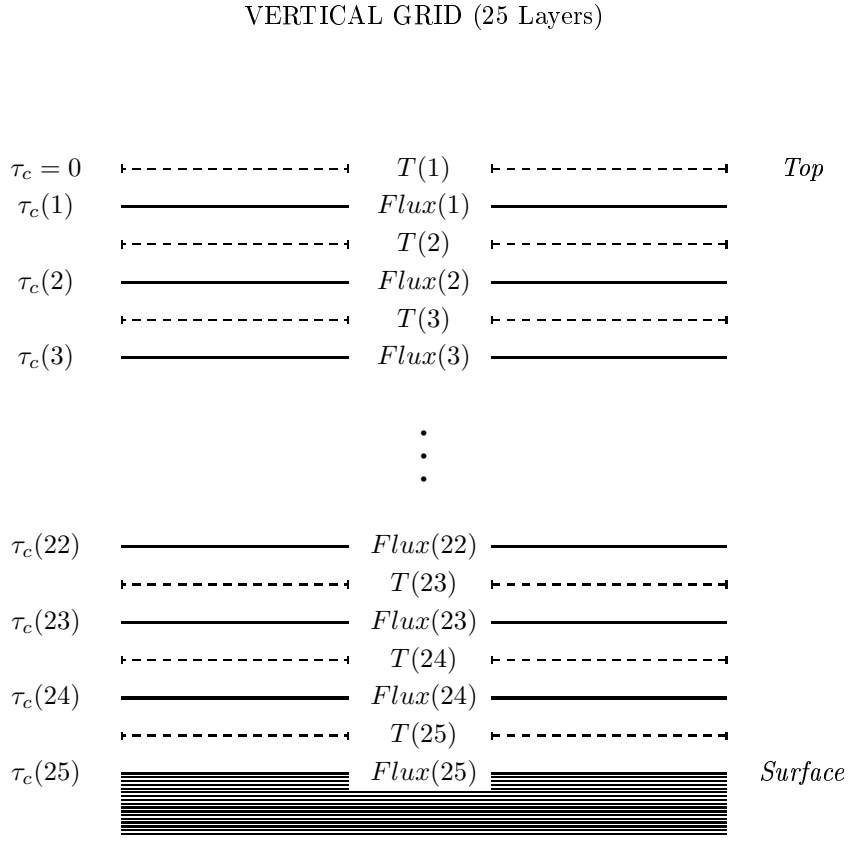

Figure 4.1: Vertical grid, 25 layer model. Flux(L) is the flux passing through layer L. $\tau_c(L)$ is the cumulative optical depth, measured from the top of the atmosphere to the bottom of layer L. T(L) is the temperature of layer L. Note that the top and bottom layers are different from the rest; the temperature is carried at the top of layer one, and the ground temperature is carried at the surface.

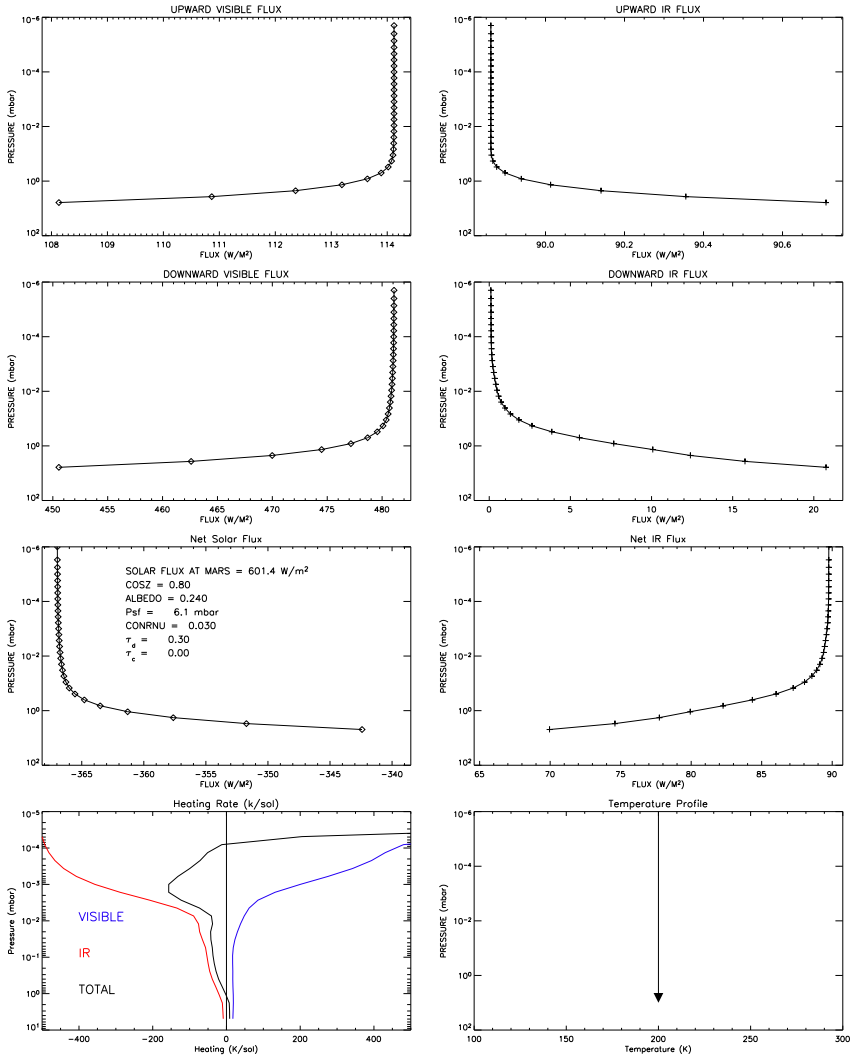

Figure 4.2: Output from the baseline driver.f configuration.